تصميم وتنفيذ نظام حماية منزل بشكل كامل

Design and implementation of the house protection system fully

٢٠١٦/٢٠١٧

تأليف وتصميم : المهندس عمار محمد هلال

هاتف : ٠٠٩٦٣٩٣٥٢٠٨٨٨٠٥

البريد الالكتروني : ammar1992888@Gmail.com

بسم الله الرحمن الرحيم

﴿وقل ربي زدني علما"﴾

صدق الله العظيم

الى من غمرني بواسع رحمته . . وأنعمني بكثير من كرمه . . . وحماني بجليل سـلطانه . . .

خالقي عز وجل

فهرس المشروع

فكرة المشروع:

تصميم مجموعة من الدارات الالكترونية التي تتحكم بأجهزة المنزل وذلك من حيث الحرارة والاضاءة وأمن المنشأة عن طريق ربط هذه الدارات بالمايكروكونترولر وتصميم برنامج محاكاة باستخدام لغة البرمجة Delphi7.

❖ الدارات والعناصر المستخدمة في المشروع:

١- متحكم صغري MICROCONTROLORمن نوع PIC .

٢- دارة تغذية .

٣- دارة ملائمة MAX232.

٤- منفذ COM في الحاسوب .

٥- برنامج التحكم في الحاسب.

٦- حساس الحرارة LM35

٧- حساس الانارة.

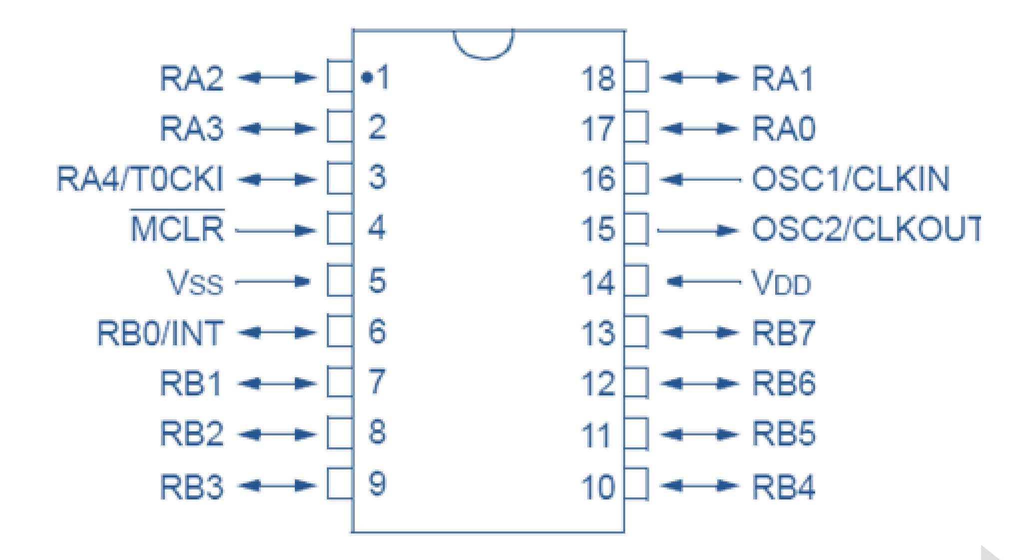
٨- مرسلات ومستقبلات الأشعة تحت الحمراء.

٩- محرك مستمر DC Motor.

❖ المتحكم PIC16F628 :

يعتبر المتحكم PIC16F628 أحد متحكمات PIC الشعبية ويستخدم في العديد من التطبيقات الصناعية والتجارية وتطبيقات الهواة. هذا الجهاز هو ذو 18 قطب ويمكن أن يعمل بتردد 20MHZ ويمتلك ذاكرة برنامج FLASHبسعة 14*1024و68 بايت من ذاكرة المعطيات RAM و64 بايت من ذاكرة EEPROM غير المتطايرة ومؤقتاً مراقب و13 قطب دخل/خرج ومصادر مقاطعة داخلية وخارجية ومقدرة كبيرة على تصريف وتصدير التيار.

١-١ وظائف الأقطاب (PIC16F628)



الشكل(1.1)

يظهر الشكل (1.1) توضع أقطاب المتحكم PIC16F628.

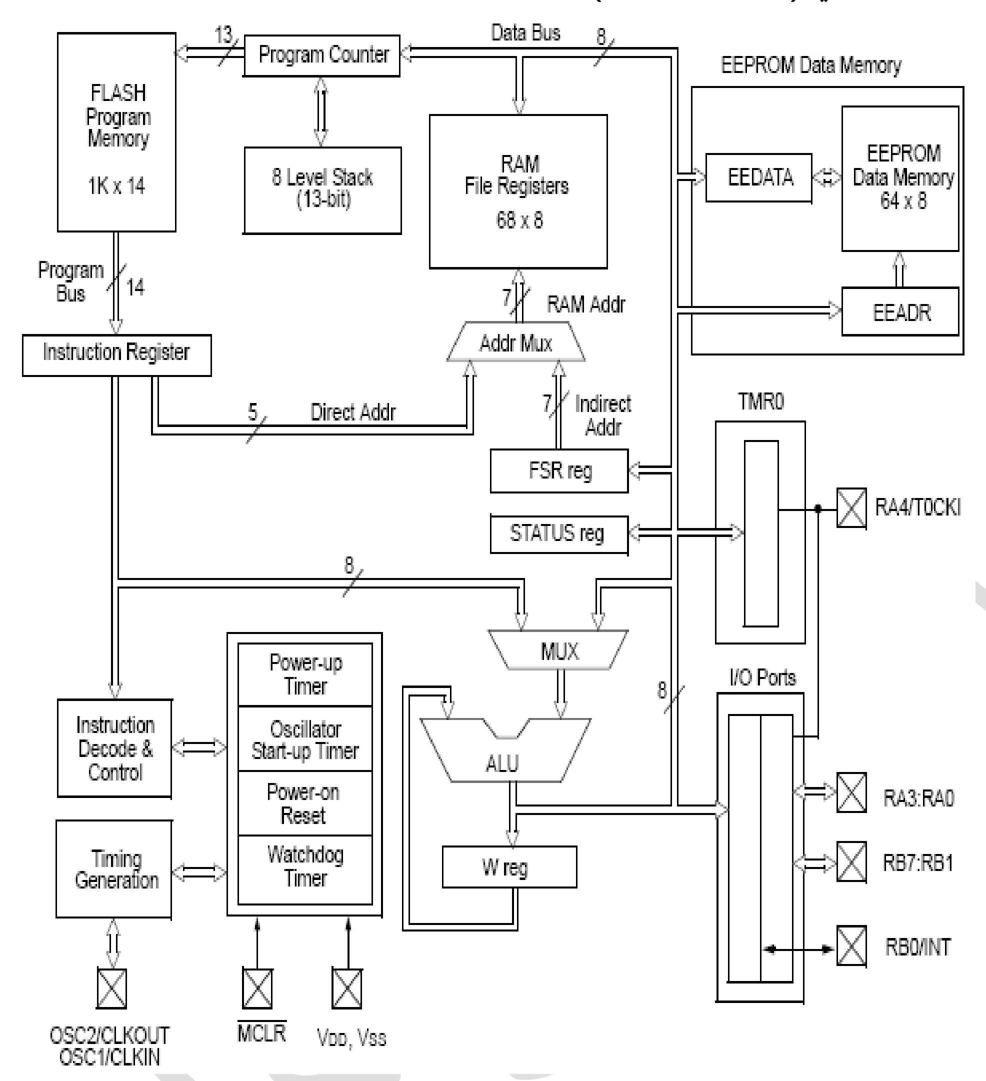
وظائف الأقطاب المتنوعة هي كما يلي:

RB0-RB7	أقطاب المنفذ B ثنائية الاتجاه
RA0-RA4	أقطاب المنفذ A ثنائية الاتجاه
VDD	جهد التغذية
VSS	أرضي
OSC1	دخل الكريستالة، المذبذب أو نبضات ساعة خارجية
OSC2	مدخل كريستالة أو المذبذب
MCLR	مدخل التصفير
INT	مدخل المقاطعة الخارجية (مشترك مع RB0)
TOCK1	مدخل اختياري لنبضات ساعة العداد (مشترك مع RA3)

الجدول (1.2)

لاحظ أن أسماء بعض الأقطاب لها خط فوقها على سبيل المثال $\overline{MCLR}$ في الشكل(1.2) هذا يعني أن هذا القطب سيكون فعالاً عندما تكون الإشارة المطبقة في المستوي المنخفض (0 منطقي في هذه الحالة).

❖ المخطط الصندوقي (PIC16F628) :



الشكل (1.3)

١-٢ مقاطعات (PIC16F628) :

يؤمّن المتحكم PIC16F628 أربعة مصادر مقاطعة داخلية وخارجية:

- مقاطعة خارجية على القطب INT.
- مقاطعة تزايد العداد .
- مقاطعة تغيير حالة الخانات الأربع العليا من المنفذ B (RB4-RB7) .
- مقاطعة إتمام عملية الكتابة في الذاكرة EEPROM .

١-٣ ذاكرة المعطيات (PIC16F628) :

تعرف الذاكرة RAM أيضاً بملف المسجلات RFM (Register File Map) وتحتوي على 80 بايتو كما يقسم ملف المسجلات إلى قسمين :مسجلات الوظائف الخاصةSFR ومسجلات الأغراض العامة GPR.يصنف ملف المسجلاتRFM كبنكين (ربما تحتوي متحكمات PIC الأكثر تعقيداً على مزيد من البنوك) البنك 0والبنك1.

ويجب اختيار بنك المسجل قبل أن تتم الكتابة أو القراءة من هذا المسجل. بعض المسجلات مشتركة بين البنكين. إن مسجلات الوظائف الخاصة SFR هي مجموعة من المسجلات تستخدم من قبل CPU للتحكم بالعمليات الداخلية والطرفيات مثل ضبط اتجاه الدخل/الخرج، إرسال معطيات إلى منفذ دخل/خرج، تحميل مسجل العداد. مسجلات الوظائف الخاصة المستخدمة خلال برمجة المتحكم باستخدام لغة عالية المستوى مشروحة في الفقرة التالية.

File Address	Bank 0	Bank 1	File Address
00h	Indirect addr.(1)	Indirect addr.(1)	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2(1)	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch – 4Fh	68 General Purpose Registers (SRAM)	Mapped (accesses) in Bank 0	8Ch – CFh
50h – 7Fh			D0h – FFh

الشكل (1.3) ملف مسجلات المتحكم PIC16F628

١-٤ مسجلات الوظائف الخاصة(PIC16F628)

نبين بالشكل (1.4) ملف المسجلات الخاصة.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page
Bank 0											
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- ----	11
01h	TMR0	8-bit Real-Time Clock/Counter								xxxx xxxx	20
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11
03h	STATUS[2]	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11
05h	PORTA[4]	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	16
06h	PORTB[5]	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	18
07h	—	Unimplemented location, read as '0'								—	—
08h	EEDATA	EEPROM Data Register								xxxx xxxx	13,14
09h	EEADR	EEPROM Address Register								xxxx xxxx	13,14
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC[1]					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10
Bank 1											
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
83h	STATUS [2]	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	16
86h	TRISB	PORTB Data Direction Register								1111 1111	18
87h	—	Unimplemented location, read as '0'								—	—
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
89h	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	14
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC[1]					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10

Legend: x = unknown, u = unchanged. - = unimplemented, read as '0', q = value depends on condition

- المسجل OPTION_REG:

هذا المسجل قابل للقراءة والكتابة في العنوان 0X81 من ملف المسجلات. يتحكم هذا المسجل بمقسم التردد المؤقت، اختيار جبهة ساعة المؤقت، مصدر ساعة المؤقت، اختيار جبهة المقاطعة الخارجية ومقاومة الرفع للمنفذ B، لضبط قطب المقاطعة الخارجية حيث أن:

الخانة 7: التحكم بمقاومات الرفع للمنفذ B

1: مقاومات الرفع معطلة.

0: مقاومات الرفع ممكّنة.

الخانة 6: اختيار جبهة المقاطعة الخارجية INT

1: المقاطعة عند الجبهة الصاعدة للقطب INT

0: المقاطعة عند الجبهة الهابطة للقطب INT

الخانة 5: مصدر ساعة العداد TMR0

1: نبضات خارجية على القطب T0CK1

0: ساعة داخلية

الخانة 4: اختيار جبهة المصدر للعداد TMR0

1: الزيادة عند الانتقال مرتفع – منخفض للقطب T0CK1

0: الزيادة عند الانتقال منخفض – مرتفع للقطبT0CK1

الخانة 3: تخصيص مقسم التردد

1: مقسم التردد مخصص لمؤقت المراقب

0: مقسم التردد مخصص للعداد TMR0

الخانات2:معدل تقسيم التردد:

000 1:2 001 1:4 010 1:8 011 1:16 100 1:32

101 1:64 110 1:128 111 1:256

- المسجل INTCON :

هذا هو مسجل التحكم بالمقاطعة المتوضع في العنوانين 0X0B و0X8Bمن ملف المسجلات. هذا المسجل مستخدم لتفعيل/تعطيل المصادر المتعددة للمقاطعة وأعلامها حيث أن:

الخانة 7: التحكم العام بالمقاطعة

1: تمكين مل المقاطعات غير المحجوبة

0: تعطيل كل المقاطعات

الخانة 6: مقاطعة اكتمال الكتابة على الذاكرة EEPROM

1: تمكين مقاطعة انتهاء الكتابة على الذاكرة EEPROM

0: تعطيل مقاطعة انتهاء الكتابة على الذاكرة EEPROM

الخانة 5: مقاطعة تزايد العداد

1: تمكين مقاطعة العداد

0:تعطيل مقاطعة العداد

الخانة 4: التحكم بالمقاطعة الخارجية INT

1: تمكين المقاطعة الخارجية INT

0: تعطيل المقاطعة الخارجية INT

الخانة3: التحكم بمقاطعة تغير الأقطابRB7-RB4

1: تمكين مقاطعة تغيير حالة الأقطاب RB7-RB4

0: تعطيل مقاطعة تغيير حالة الأقطاب RB7-RB4

الخانة 1: علم المقاطعة الخارجية INT

1: حدثت المقاطعة الخارجية INT

0: لم تحدث المقاطعة الخارجية INT

الخانة 0: علم مقاطعة تغير الأقطاب RB7-RB4

1: واحد أو أكثر من الأقطاب RB7-RB4 تم تغيير حالته

0: لم يتم تغيير حالة أي قطب من الأقطاب RB7-RB4

١-٥ منافذ I/O(PIC16F628) :

- المنفذ PORTA:

إن المنفذ A هو بعرض خمس خانات ثنائية وأقطابه هي RA0 – RA4 وهو متوضع في العنوان 5 من ملف المسجلات. الخانات الأربعة الدنيا RA3 – RA0 تمتلك دارات قيادة خرج CMOS مع مقدرة على تصريف وتصدير تيار حتى 25Ma. القطب RA4ذو مصرف مفتوح، ويجب وصله مع مقاومة رفع مناسبة عند استخدامه كخرج. أقطاب المنفذ A ثنائية الاتجاه ويتم تحديد اتجاهها عن طريق المسجل TRISA. فتفعيل أحد خانات هذا المسجل يجعل القطب الموافق في المنفذ A دخلاً، وبشكل مشابه فإن تصفير أحد خانات هذا المسجل يجعل القطب الموافق في المنفذ Aخرجاً. فإذا أردنا على سبيل المثال جعل الخانات0,1,2 من المنفذ A كدخل وبقية الأقطاب كخرج يجب أن نحمل في المسجل TRISA القيمة الثنائية 00000111.

- المنفذ PORTB:

هذا المنفذ هو بعرض ثماني خانات ثنائية وأقطابه هي RB0-RB7 وهو متوضع في العنوان 6 من ملف المسجلات. تمتلك الأقطاب دارة قيادة خرج من نوع CMOS وتستطيع تصريف/تصدير تيار حتى 25mA. القطبRB0يمكن أن يستخدم كقطب للمقاطعة الخارجية. وبشكل مشابه فإن الأقطاب RB4-RB7يمكن أن تستخدم لتوليد مقاطعة عندما تتغير حالة أحد هذه الأقطاب. المنفذ B ثنائي الاتجاه ويتم تحديد اتجاه أقطابه عن طريق المسجل TIRSBفتفعيل أحدى خانات هذا المسجل يجعل القطب الموافق دخلاً أما تصفيرها فيجعل القطب الموافق في المنفذ Bخرجاً (مثلاً لجعل الخانات 0,2,4 من المنفذ B كدخل وباقي الأقطاب كخرج) يجب أن نحمل المسجل TIRS B بالقيمة:00010101 .

١-٦ معالجة المقطعات (PIC16F628):

من أجل أن يتم قبول المقاطعة من قبل المعالج يجب تحقق الشروط التالية:

- أن يتم تمكين علم المقاطعة العام في المسجل INTCON (GIE=1)
- علم المقاطعة لمصدر المقاطعة في المسجل يجب أن يُفَّعل (مثلاً INTE=1 لتفعيل المقاطعات الخارجية)
- المقاطعة يجب أن تظهر فيزيائياً مثلاً (قطب INT يجب أن يرفع إلى المستوى 1 منطقي إذا تم ضبط INTEDG بشكل مسبق إلى 1 منطقي)

بعد أن يتم كشف المقاطعة سيقفز البرنامج إلى برنامج خدمة المقاطعة المتوضع في العنوان 4 من ذاكرة البرنامج. في هذه المرحلة يتم تعطيل المقاطعات الأخرى ويجب تصفير علم المقاطعة الحاصلة (مثل الخانة INTF للمقاطعات الخارجية INTCON) من أجل قبول مقاطعة جديدة من مصدر المقاطعة.

١-٦-١المقطعات الخارجية INT(PIC16F628) :

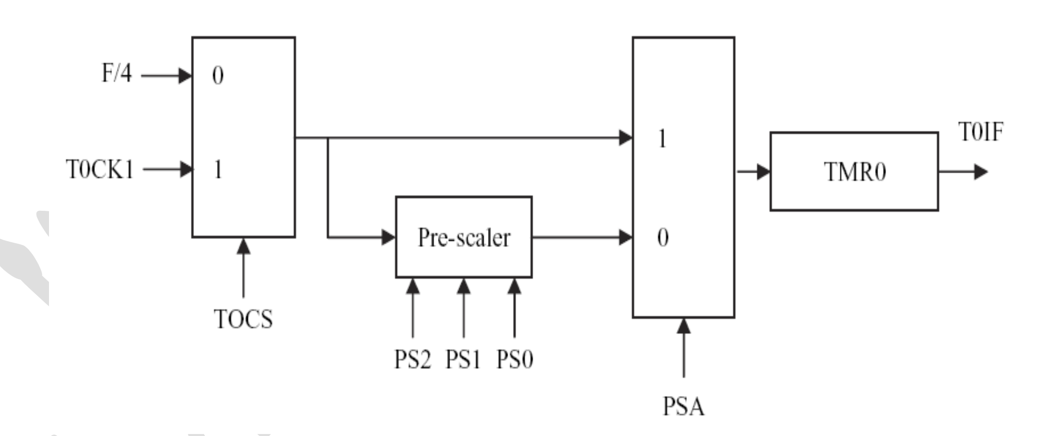
هي الاستجابة لحدث الجبهة الصاعدة أو الهابطة على قطب المقاطعة الخارجية (int)حيث يتم إعداد حدث المقاطعة عن طريق الخانة 6

1: المقاطعة عند الجبهة الصاعدة للقطب INT

0: المقاطعة عند الجبهة الهابطة للقطب INT

١-٧ المؤقت TMR0(PIC16F628):

يتوافر في المتحكم PIC16F628عداد ذي 8 خانات. يدعى هذا العداد TMR0 ويمكن استخدامه كمؤقت أو كعداد.



الشكل (1.5) بنية العداد TMRO

يوضح الشكل (1.5) بنية هذا العداد. يتم زيادة المسجل بمقدار 1 كلما تم تطبيق نبضة ساعة على القطب الخارجي T0CK1 للمتحكم وذلك إذا تم استخدامه كعداد.

أما عندما يتم استخدامه كمؤقت تزداد قيمة المسجل بمعدل يتم تحديده بتردد ساعة النظام ومقسم التردد الذي يتم اختياره عن طريق المسجل OPTION-REG.تتراوح قيم تقسيم التردد مابين1:2 إلى 1:256، فعلى سبيل المثال عند استخدام ساعة بتردد 4MHZ فإن دورة التعليمة هي 1µsec (الساعة ذات التردد 4MHZ

يكون دورها 0.25µsec لكن الساعة تقسم داخليا على 4 للحصول على الدورة الأساسية للتعليمة). إذا اخترنا معدل تقسيم تردد 1:8 فستتم زيادة قيمة مسجل العداد كل8µsec.

يتم توليد مقاطعة تزايد العداد عندما يتزايد مسجل العداد من 255 إلى الصفر. هذه المقاطعة يتم تمكينها برمجياً عن طريق تفعيل الخانة 5 من مسجل INTCON. فإذا أردنا على سبيل المثال توليد مقاطعة كل 200µsec عند العمل بتردد 4MHz فيجب علينا اختيار مقسم التردد 1:4 وتفعيل مقاطعة تزايد العداد. عندها ستكون القيمة الفعالة لمعدل ساعة العداد 4µsec، ومن أجل فترة 200µsec يجب إرسال 200/4=50 نبضة ساعة إلى مسجل العداد، ولذلك يجب تحميل مسجل العداد بالقيمة 256-50=206

١-٨ ذاكرة EEPROM(PIC16F628) :

يمتلك المتحكم PIC16F62864 بايت من ذاكرة EEPROM غير المتطايرة والتي يتم التحكم بها عن طريق المسجلات EEDATA, EEADR, EECON1, EECON2 وهناك تعليمات لقراءة أو كتابة البيانات على هذه الذاكرة.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other RESETS
08h	EEDATA	EEPROM Data Register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM Address Register								xxxx xxxx	uuuu uuuu
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM Control Register 2								---- ----	---- ----

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends upon condition.

الشكل (1.6)

١-٩ المسجل الضبط CONFIGURATION:

يحتوي المتحكم PIC16F628 أيضاُ على مسجل ضبط، و يمكن تفعيل أو تصفير خاناته أثناء البرمجة. يحتوي هذا المسجل على خانات من أجل اختيار نمط الهزاز وتمكين/تعطيل حماية الشيفرة وتمكين/تعطيل عداد التشغيل وتمكين/تعطيل مؤقت المراقب.

١- دارة التغذية:

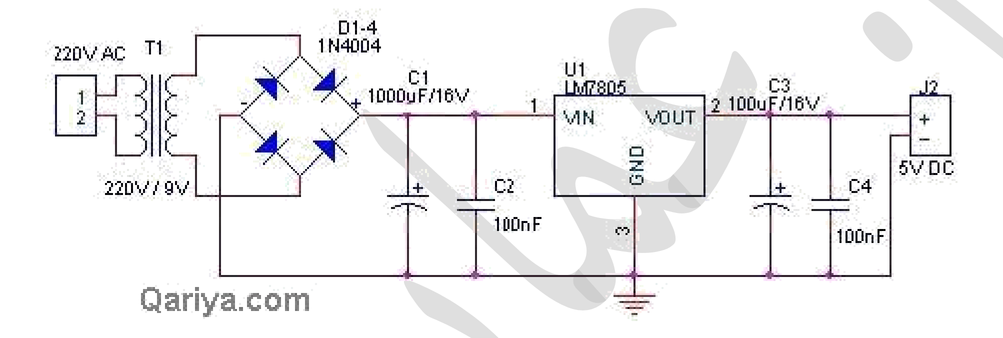
- وصف الدائرة:

تعتبر هذه الدائرة ابسط و أفضل دوائر التغذية المنظمة للجهد المستمر .. ولهذا فهي شائعة جدا .. وبالإضافة إلى بساطة تركيبها .. هي أيضا تتكون من عناصر رخيصة ومتوفرة

- مواصفات الدائرة:

– مصدر التغذية ٢٢٠ فولت أو ١١٠ حسب المحول
– جهد المخرج : ٥ فولت جهد مستمر منظم
– تيار الخرج : ١ أمبير
– الدائرة المتكاملة تحتوي على حماية داخلية ضد القصر Short Circuit

- مخطط الدائرة:



- شرح الدائرة :

في البداية يقوم المحول الكهربائي بتخفيض جهد المصدر سواء كان ١١٠ فولت أو ٢٢٠ فولت إلى جهد صغير متردد 12فولت

ثم تأتي قنطرة التوحيد التي تقوم بتحويل موجة الجهد المتردد إلى جهد مستمر ذو تموج في الخرج .. والتي يستقبلها المكثف ليقوم بتنعيم هذا الجهد للوصول قدر الامكان إلى الثبات .. ثم يأتي دور الدائرة المتكاملة والتي تحتوي في داخلها على مجموعة من الترانزستورات التي تقوم بتنظيم جهد المستمر الغير منظم لتعطي في الخرج جهد ثابت مستمر منظم..

٢- الربط مع الحاسب (MAX232):

▪ التعامل مع المنفذ التسلسلي COM

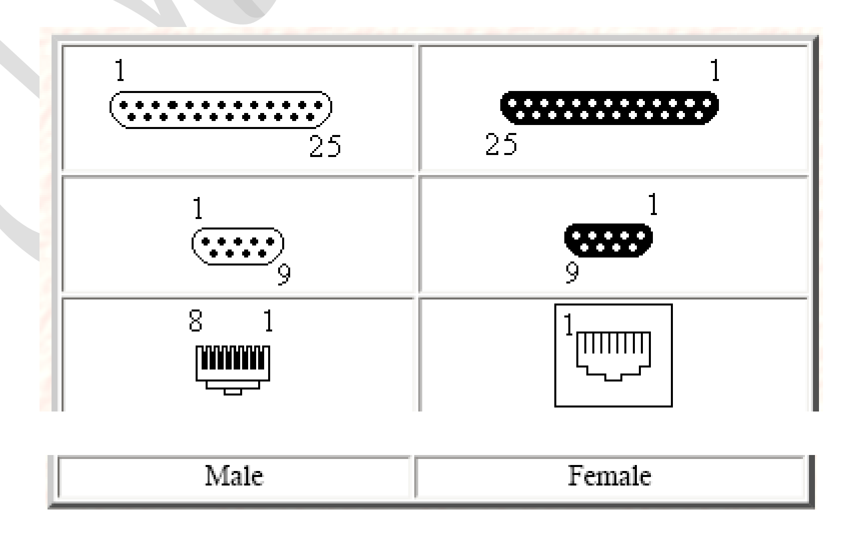
مقدمة:

يختلف المنفذ التسلسلي عن المنفذ التفرعي في طريقة إرسال المعلومات حيث ترسل المعلومات وتستقبل على خط واحد بشكلٍ متسلسل (بت واحد في كل مرة). بما أن أجزاء الحاسوب الداخلية تتبادل المعطيات فيما بينها بشكل تفرعي لذلك لا بد من وجود دارة تحول المعلومات من الشكل التفرعي إلى الشكل التسلسلي قبل إرسالها عبر المنفذ التسلسلي. لذلك تعتبر عملية الاتصال التسلسلي أعقد بكثير من الاتصال التفرعي. إلا أن هذا التعقيد يقابله فوائد كثيرة أهمها القدرة على إرسال المعلومات إلى مسافات أكبر بكثير من المسافات الممكنة في حالة الإرسال التفرعي. يوجد عادةً في الحواسيب الشخصية اثنين من المنافذ التسلسلية يرمز لها بـ COM1 و COM2 على الترتيب ويمكن أن يحتوي حتى أربعة منافذ تسلسلية (COM3 و COM4) .

يعتبر المنفذ التسلسلي أداة نقل ثنائي الاتجاه (full duplex) والمقصود بذلك بأنها قادرة على استقبال وإرسال البيانات في وقت واحد ويعود ذلك إلى وجود خطين منفصلين يستخدم أحدهما من أجل الإرسال و الآخر من أجل الاستقبال.

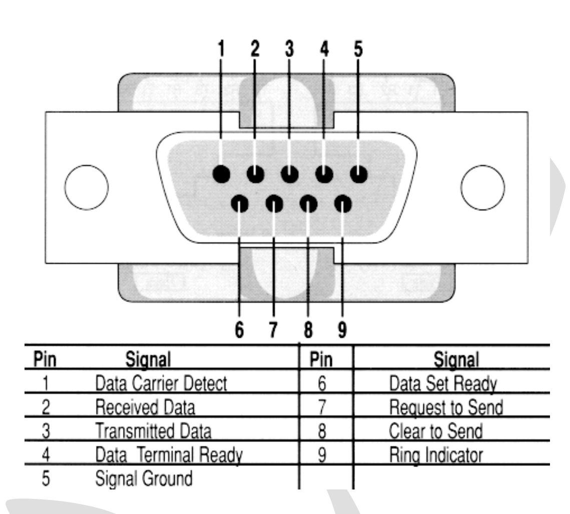
تسمى الدارة التي تقوم بالإرسال والاستقبال التسلسلي في الحاسوب الشخصي بالمرسل والمستقبل غير التزامني العام UART (أو 8250) تتصل هذه الدارة مباشرةً مع ممرات الحاسوب.

يبين الشكل التالي الأشكال المختلفة للمنفذ التسلسلي:



يستخدم الحاسوب الشخصي بروتوكول خاص بالاتصال التسلسلي يرمز له بـ RS-232 . يعرف هذا البروتوكول شكل المنفذ التسلسلي (المواصفات الميكانيكية) وجهود وترتيب إشارات المعلومات والمصافحة (المواصفات الكهربائية).

يوجد نوعين من المنافذ التسلسلية ، الأول يتألف من ٢٥ قطب وهو الشكل الأصلي المعرف في البروتوكول ويدعى D25 (قديم جداً). والثاني صغير فيه ٩ أقطاب ويدعى D9 وقد ظهر في الحواسيب الشخصية الحالية نظراً لعدم استخدام الأقطاب الأخرى في المنفذ الكبير. يبين الشكل التالي المنفذ التسلسلي D9 وتوزيع الأقطاب فيه:



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

٣- أقطاب المنفذ التسلسلي:

Description	Signal	9-pin DTE
Carrier Detect	CD	1
Receive Data	RD	2
Transmit Data	TD	3
Data Terminal Ready	DTR	4
Signal Ground	SG	5
Data Set Ready	DSR	6
Request to Send	RTS	7
Clear to Send	CTS	8
Ring Indicator	RI	9

- GND أو SG (الأرضي): يؤمن أرضي مشترك لدارتي الإرسال والاستقبال.
- TxD (Transmitted Data) المعلومات المرسلة (خرج): يتم إرسال المعلومات على هذا القطب.
- RxD (Received Data) المعلومات المستقبلة (دخل): يتم استقبال المعلومات على هذا القطب.

- DTR (Data Terminal Ready) (خرج): خط مصافحة يدل على أن طرفية المعطيات (الحاسوب) جاهزة للإرسال.
- DSR(Data Set Ready) (دخل): خط مصافحة يدل على أن جهاز المعطيات (المودم) جاهز لاستقبال المعلومات من الحاسوب.
- RTS(Request To Send) (خرج): خط مصافحة يطلب من جهاز المعطيات القيام بعملية إرسال.
- CTS (Clear To Send) (دخل): خط مصافحة يخبر الطرفية (الحاسوب) عن إمكانية بدء عملية إرسال.
- CD (Carrier Detected) (دخل): يخبر الحاسوب عن توفر إشارة الحامل (نغمة التون).
- RI (Ring Indicator) (دخل): يخبر الحاسوب عن ورود مكالمة.

تستخدم الإشارتان الأخيرتان فقط في حالة وصل مودم .

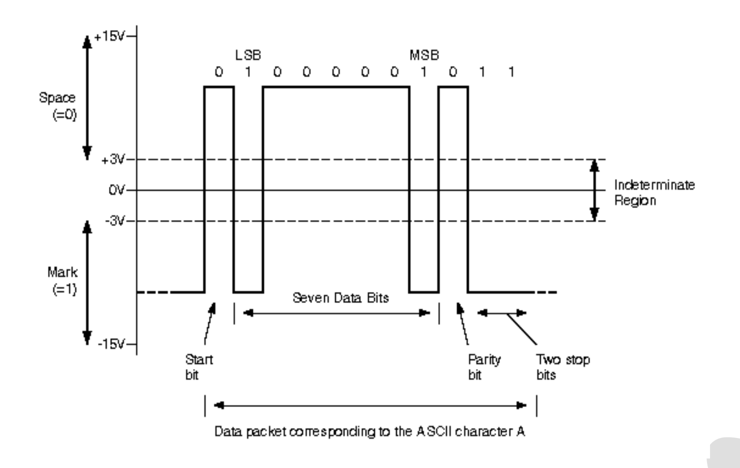
يمكن تحقيق الحد الأدنى من الاتصال بدون مصافحة باستخدام ثلاثة خطوط فقط هي SG و TxD وRxD.

- الإشارات الكهربائية:

يرمز للمستوى المنطقي ١ (علامةMark) بجهد يتراوح بين 15- ,5- فولت في المخارج و بين 13- ,3- فولت في المداخل . ويرمز للمستوى المنطقي ٠ (فراغ Space) بجهد يتراوح بين ٥ و ١٥ فولت في المخارج و ٣ و١٥ في المداخل . بما أن هذه الجهود تختلف عن الجهود المنطقية داخل الحاسوب (٠ و ٥ فولت) لذلك توجد دارات خاصة لتحويل المستويات المنطقية العادية إلى إشارات متوافقة مع RS-232 وبالعكس. توضع هذه الدارات بين UART وأقطاب المنفذ التسلسلي.

- إرسال البيانات تسلسلياً:

يتم إرسال البيانات بشكل غير متزامن وعلى شكل إطارات يحتوي كل إطار على بايت كامل (٨ بت) بالإضافة إلى بتات تحكم إضافية.



يتألف الإطار من :

- بت بداية Start Bit: يبتدئ الإطار بهذا البت لتنبيه الجهاز المستقبل بوصول المعطيات ومزامنة الآلية التي تفصل البتات عن بعضها. ويكون هذا البت عبارة عن فراغ Space وذلك لأن الخط عند عدم إرسال شيء يكون في حالة العلامة Mark .
- بتات المعطيات Data Bits: تمثل المعلومات المرسلة وتتألف من 8 bit في الحالة العادية ويمكن أن تكون 7 bitفي حال إرسال رسائل نصية بترميز آسكي (حيث يمكن الاستغناء عن البت الأخير).
- بت الازدواجية Parity Bit: تستخدم عملية فحص الازدواجية للتحقق من صحة الإرسال. يضاف بت يدعى بت الازدواجية إلى الإطار . تعتمد قيمة هذا البت على محتويات بتات المعطيات وتحسب بعدة طرق. الازدواجية الزوجية تعني أن عدد العلامات (الواحدات المنطقية) في بتات المعطيات بما فيها بت الازدواجية زوجي. أما الازدواجية الفردية فهي تحسب بحيث يكون عدد العلامات فردي.
- بتات التوقف Stop Bits: يوجد في نهاية كل إطار عدد من بتات التوقف ويمكن أن تكون بت واحد أو بت ونصف أو بتان. وتكون بتات التوقف في حالة Mark وذلك لتمييز الإطار عن الإطار التالي الذي سيبتدئ حتماً بعلامة Space تمثل بت البداية .

يمكن إرسال و استقبال المعطيات عبر المنفذ التسلسلي بعدة طرق . يمكن استخدام مقاطعةBIOS ذات الرقم 14H للقيام بذلك وتعتبر الطريقة الأسهل أو يمكن برمجة UART بشكل مباشر وهي الطريقة التي تؤمن المرونة الكاملة. أما في نظام Windows فيجب استخدام وظائف API الخاصة بالاتصالات التسلسلية، كما يمكن استخدام Hyper Terminal، وكذلك Matlab للتعامل مع المنفذ التسلسلي.

❖ لوحة ملائمة المنفذ التسلسلي S-board V2.0:

سنتحدث في هذا القسم عن لوحة القيادة المستخدمة للوصل مع المنفذ التسلسلي، وعن الطريقة البرمجية في التعامل معها، بالإضافة إلى تطبيق أمثلة عملية عليها.

- مزايا اللوحة S-board:

- متلائمة مع المنفذ التسلسلي RS-232.
- برنامج تشغيل خاص بـ RS-232متوضع على اللوحة.
- دارة UART من خلال متحكم خاص UTX8100.
- اختيار معدل النقل Baud rate وصيغة البيانات من خلال وصلات.
- إعلام عن الحالة والأخطاء من خلال لدات ضوئية.
- تحوي دارة تحويل للناقل I2C.
- منافذ (وصلات أرجل) لوصل خطوط البيانات DATA BUS، ودخل UART و خرجUART و Shift OUT و S-bus للوصل مع لوحات EX الأخرى.

- شرح اللوحة S-board:

- وصلة S-bus: يتم إرسال كافة إشارات المنفذ التسلسلي عبر الدارات MAX232، إلى بفر (ذواكر مؤقتة). تستخدم الدارة 74HC541 لقيادة التيار وحماية المنفذ من الأخطاء.
- دارة Max 232: تستخدم لتحويل الجهود من 12v-إلى 5v ومن 12v+ إلى 0v وبالعكس. ويمكن وصل منفذين تسلسليين إليها.
- وصلة الخرج التسلسلي SERIAL OUTPUT: إشارات خرج المنفذ التسلسلي هي TxD, DTR, RTSوهي موصولة إلى وصلة SERIAL OUTPUT لقيادة أجهزة الخرج.
- وصلة الدخل التسلسلي SERIAL INPUT: إشارات دخل المنفذ التسلسلي هي RxD, DSR, CTS, DCDوهي موصولة إلى وصلة SERIAL INPUT لاستقبال إشارات الدخل الرقمية.
- وصلة I2C: يتضمن الناقل I2Cخطين ثنائيين SDA و SCL، وبما أن خطوط المنفذ التسلسلي ليست ثنائية الاتجاه فنحن بحاجة إلى ٣ خطوط لتوليد إشارة الناقل I2C. من أجل SDA يتم استخدام RTS للإرسال و CD للاستقبال. ومن أجل SCL يتم استخدام DTR لقيادة نبضات الساعة من خلال الترانزستور Q2.
- الدارة UART: قلب هذه الدارة هو المتحكم UTX8100 UART، وعمله يشبه عمل الشريحة UART على اللوحة الأم في الحاسوب. يعمل UTX8100 كملائمة مع المنفذ التسلسلي. إشارات البيانات هي TxD, RxD.

يرسل TxD البيانات التسلسلية إلى UTX8100 من خلال RxD1,RxD2 . يقوم بعد ذلك UTX8100بتحويلها إلى بيانات تفرعية 8 bit تخرج على D0-D7. ثم تمرر جميع إشارات الخرج إلى دارة المسك (البفر) 74HC541، وخرج الدارة إلى وصلات UART OUT من أجل الوصل مع أجهزة الخرج.

يتقبل RxD البيانات التسلسلية من TxOut للدارة UTX8100 عندما تقدح بالمنطق 0 من DTR عند REQ في UTX8100. بيانات الدخل تؤخذ من D0-D7 ويتم مسكها من وصلة UART IN بواسطة 74HC541. حيث يقوم UTX8100 بتحويل البيانات التفرعية 8 bit إلى تسلسلية ويرسلها عبر RxD.

يستطيع UTX8100 اختيار معدل الإرسال Baud rate وفق ٨ قيم:

1200, 2400, 4800, 9600, 14400, 19200,28800 and 57600 bit per second

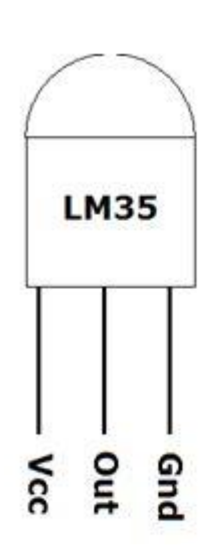
من خلال الوصلة JP1 الموصولة مع BAUD1-BAUD3 للمتحكم UTX8100.

يستطيع أيضاً UTX8100التحكم بحجم البيانات المرسلة في الإطار وخانة الازدواجية وخانات التوقف من خلال وصلات. كما يستطيع كشف أخطاء الازدواجية PE والإطار FE التي تعمل في المنطق 0 ويمكن وصلها دارة اظهار تعمل بالمنطق ٠ لرؤية العملية. وفي حالة عدم وجود بيانات في مسجل الإرسال للمتحكم فسيكون READY في وضعية ١ وهذا يدل على جاهزية الإرسال.

❖ وصلة التزامن SHIFT OUT:

تستطيع اللوحة توسيع عمل منفذ الخرج للاتصال ببيانات متزامنة من خلال SHIFT OUT. فهو يتضمن الاشارات SHIFT CLK, SHIFT DATA من خرج المتحكم UTX8100، حيث ترسل إشارات إلى وصلة SHIFT OUT باسم CLK (clock), and D0 (serial data). تستقبل LE(Latch enable) إشارة منRTS، وRESET (RST) هو SHIFT RST للمتحكم.

٥- حساس الحرارة LM35



هذا الحساس له ثلاثة أطراف كما في الشكل . جهد التغذية Vcc يمكن أن يكون من 4Vإلى 20V . لاستخدام الحساس ببساطة قم بتوصيل الطرف Vcc بالجهد +5Vوطرف الأرضي (المشترك) GND بالأرضي وطرف الخرج Outبأحد أطراف المداخل التناظرية للميكروكونترولر . هذا الخرج يتغير خطيا مع درجة الحرارة وهو يساوى ١٠ ملى فولت لكل درجة مئوية .

إذا كان الخرج , 310 mVعلى سبيل المثال ، فإن درجة الحرارة تكون ٣١ درجة مئوية . المحول ADC بالميكروكونترولر PICبسعة ١٠ بت ،إذا كان جهد المرجع المستخدم هو 5Vفإن دقة التحويل (أقل جهد يمكن قياسه) تكون:

5/1024 = 5mV approx

بمعنى آخر : أقل نتيجة تحويل ADCهي البت أو الوحدة , "1"عندئذ يكون أقل جهد يمكن قياسه هو 5mV وهو يناظر درجة حرارة تساوى 0.5 درجة مئوية وهى دقة القياس.

ملاحظة:

القيمة الدقيقة هي 4.88mv .

مثال :

إذا كانت نتيجة التحويل ADCهي 10فهذا يعنى أن جهد الدخل:

10 x 5mV = 50mV

وهو يناظر ٥ درجة مئوية (طل درجة مئوية تناظر ١٠ ملى فولت.)

يمكننا قراءة قيمة أي قناة تحويل ADC باستخدام الدالة " ADC_Read(ch)" حيث ch هي رقم القناة .

إذا قمت بتوصيل الحساس LM35إلى القناة RA0/AN0عندئذ تصبح الدالة . " ADC_Read(0)" هذه القراءة يجب حفظها في متغير ليصبح شكل العبارة :

result = ADC_Read(0);

هذه العبارة تعنى تخزين القراءة الحالية لنتيجة التحويل ADCفي المتغير . resultبيانات هذا المتغير يجب أن تكون من نوع "(int) unsigned "65535 .. 0حيث أن مدى قيمة نتيجة التحويل هو . 0-1023

كما ذكرنا : قيمة نتيجة التحويل ADCتكون نتيجة "لمضاعفات" القيمة 5mVونظرا لأن كل واحد درجة مئوية

تنتج 10mV في خرج الحساس ، نتيجة لذلك يمكننا القول " كل وحدتين من وحدات نتيجة التحويل ADC تساوي واحد درجة مئوية" . لذلك ، للحصول على درجة الحرارة يتم قسمة قيمة المتغير resultعلى 2أي:

result = result /2;

ملاحظة:

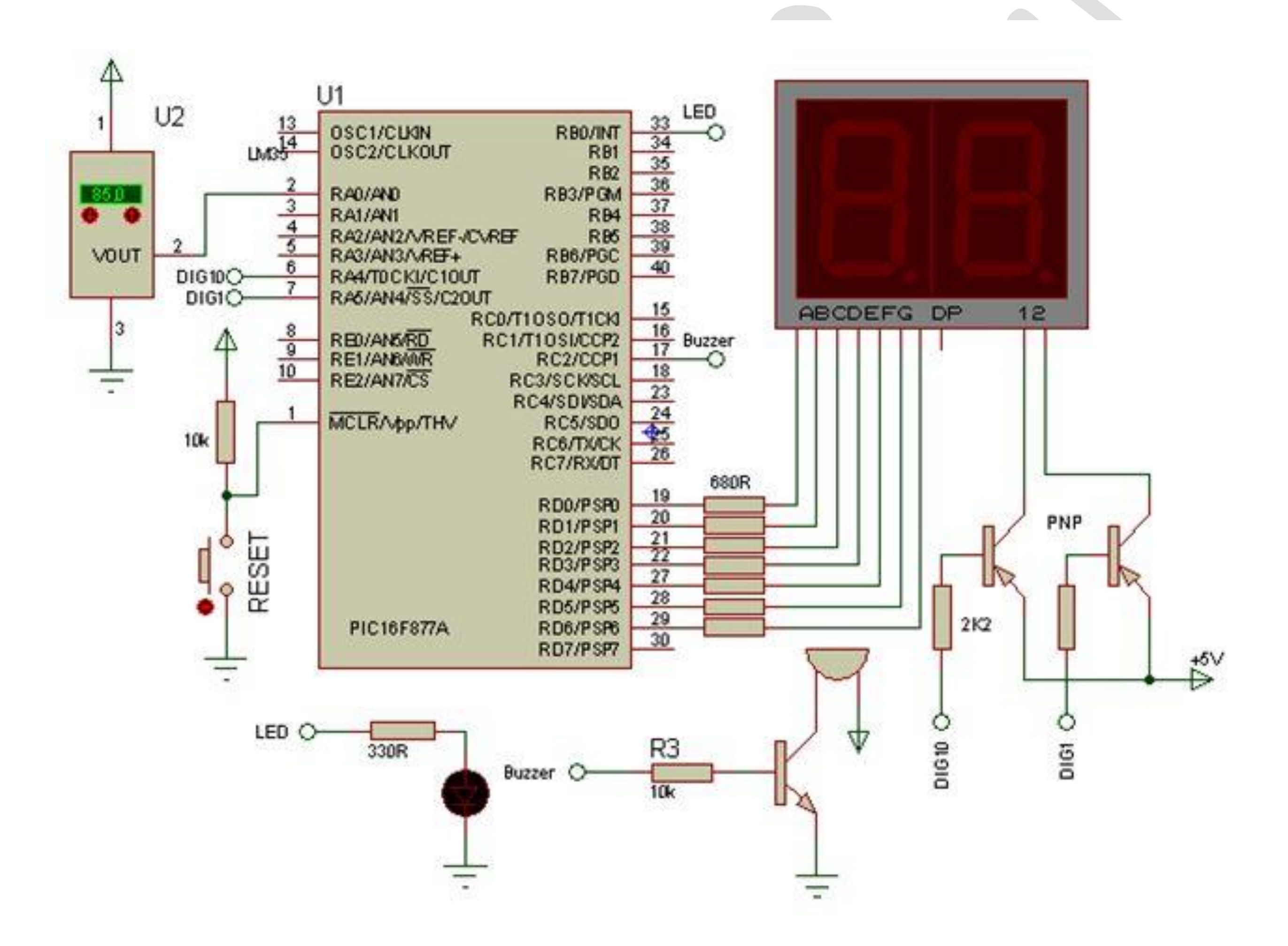
معامل القسمة الدقيق هو , 2.05أي :

result = result /2.05;

في النهاية يمكنك عرض هذه القيمة على وحدات العرض المختلفة.

في هذا المشروع سوف نستخدم وحدتين 7 segmentلعرض درجة الحرارة

❖ الدارة الالكترونية:



❖ مبدأ العمل:

يتم تحديد درجة الحرارة المناسبة في المصنع بشكل الي عن طريق المايكروكونترولر و عن طريق برنامج الدلفي على الحاسب، عند تجاوز درجة حرارة المحيط الدرجة المطلوبة يقوم المتحكم الصغري (المايكروكونترولر) بأرسال أمر الى دارة التبريد (مكيف التبريد) ليتم تشغيله حتى تصل درجة الحرارة الى الحد المطلوب فيرسل المتحكم الصغري (المايكروكونترولر) أمراً بإيقاف عمل دارة التبريد (تحكم الي بدرجة الحرارة عن طريق التغذية الخلفية البرمجية).

❖ البرنامج :

```
*/Project:

Temperature display and controller using two 7 segment and LM35

Connection as in QL200:

segments connected to PORTD-Common Anodes :RA4>tens & RA5>ones

LED : connected to RB0

Buzzer :connected to RC2

MCU : PIC16F877A

MikroC PRO

ENG.F.ABDELAZIZ

http://www.eeecb.com/vb/index.php

/*

-----------------------------------------------------------------//
------------------------------------------

//define the table of constant 0-9 (mask (

const char
TABLE[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90;{
```

```
unsignedint  result=0x00;                    //define ADC convert result register

unsigned short i;                    // for loop counter


void  init();                    //I/O PORT initialize function declare

void  display(int result);                    //display function declare


------------------------------------------------------------------//
------------------------------------------------

// main program

void  main()

}

init();                    //call initialize function

while(1          (

}

result=0x00;                    //clear the convert result

result = ADC_Read(0);// read and storADCresult as "0-1023"

//   result=result/2;  // covert from "0-1023" to "temperature in degree

result=result/2.05;  // exact conversion from "0-1023" to "temperature in degree

display(result);                    //call display function

 //Temperature controller part
```

```
if(result>75) PORTB.B0=1; else PORTB.B0=0; // LED

if(result>85) PORTC.B2=1; else PORTC.B2=0;//Buzzer

{

{

------------------------------------------------------------//

// initialize function


void  init()

}

ADCON1=0X8E;              //set RA0 pin > ADC input ,the others general I/O

TRISA=0X01;               //set RA0 INPUT,the others OUTPUT

TRISD=0X00;                  //set D PORT all OUTPUT

TRISB.B0=0;                   //For LED

TRISC.B2=0;                  //For Buzzer

PORTD=0XFF;               //close all display

{

------------------------------------------------------------//

//display function

void display(int result(

}

intTEN,ONE,temp;   //define 3 temporary variable
```

```
temp=result;                                    //temporary keep AD convert result

TEN = temp/10 ;                        //get display tens bit

ONE = (temp % 10) ;                    //get display ones bit

----------------------------------------//

for (i = 0; i<=50; i} (++


PORTD=TABLE[TEN];                   //get the display tens bit code from
table

PORTA=0x2F;                            //RA4 OUTPUT low,light tens bit display

Delay_ms(1);                            //delay some time,ensure display
brightness


PORTD=TABLE[ONE];            //get the display ones bit code from table

PORTA=0x1F;                            //RA5 OUTPUT low,light ones bit display

Delay_ms(1);                            //delay some time,ensure display
brightness

{

------------------------------------------------------------//

{
```

٧- حساس الاضاءة:

المقاومة الضوئية LDR

هي مقاومة حساسة للضوء تتغير مقاومتها من حوالى ١٠٠٠٠٠٠ اوم الى لا مقاومة عند سقوط الضوء عليها ومن اجل هذه الخاصية استفاد منها الفنيون ووضعوا دوائر تستخدم خاصية هذه المقاومة

هناك دوائر كثيرة سنوردها لاحقا تستغل خاصية تاثر المقاومة بالضوء هناك دوائر انذار بالضوء وايضا انذار بالظلام.

كيف تعمل هذه المقاومة:

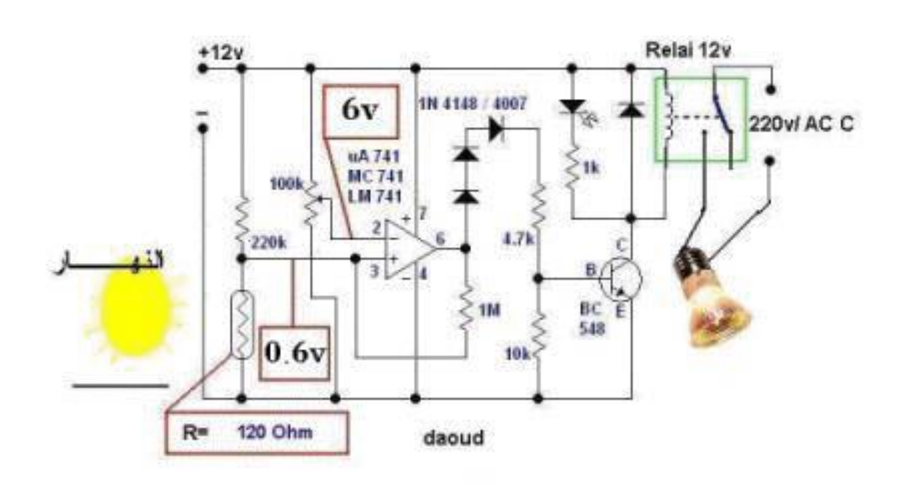
تكون المقاومة عالية وبالتالي تمنع التيار من المرور من خلالها وعندما يسقط الضوء عليها تنهار او تسقط مقاومتها الى الصفر فيمر التيار الى الجهة الاخرى وهذه هي فكرة اغلب الدوائر التي تستخدم المقاومة الضوئية

الشكل العام للمقاومة الضوئية

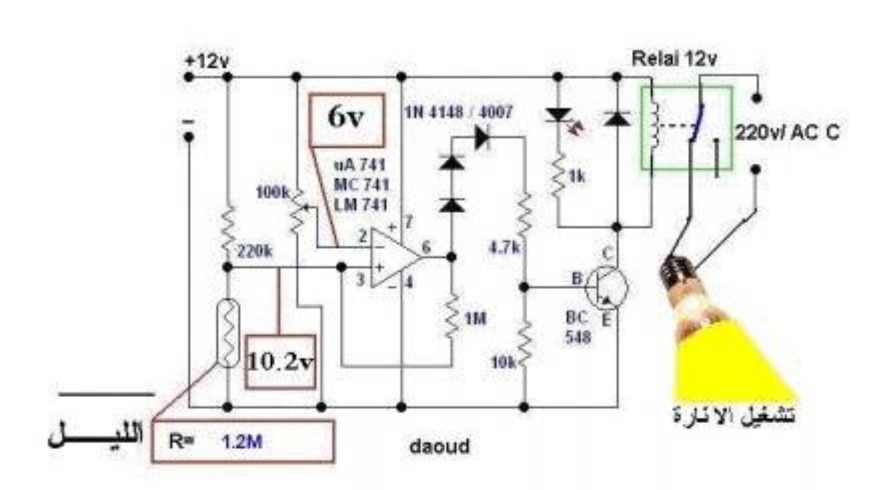


❖ الدارة المستخدمة في المشروع:

١- في النهار



٢- في الليل:

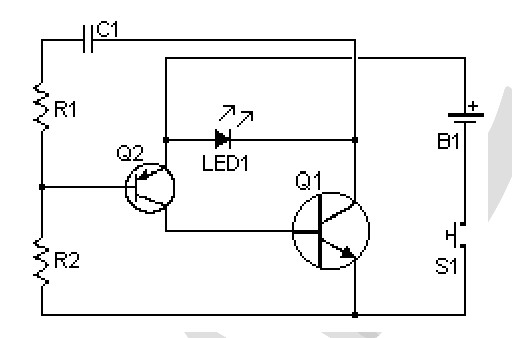


٨- حساس الاشعة تحت الحمراء:

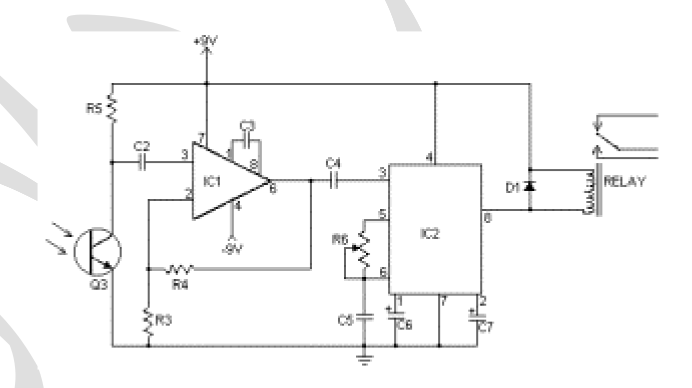
- شرح الدائرة :

- لمعايرة الدارة: نثبت المفتاح اللحظي (S1) بوضعية تشغيل الدارة ونوجه ليد الأشعة تحت الحمراء باتجاه دارة الاستقبال، ثم نقوم بتغيير قيمة المقاومة المتغيرة (R6) حتى نسمع صوت تغيير وضع تماس الحاكمة
- يمكن زيادة مجال عمل الدارة باستخدام خرج ليد أشعة تحت الحمراء أعلى.
- الانتباه إلى أن أي حاجز سيوقف استجابة دارة المستقبل للمرسل.

- مخطط دائرة المرسل :



- مخطط دائرة المستقبل :



- العناصر الإلكترونية المطلوبة لهذه الدارة :

R1 : مقاومة (4/1) واط (22)كيلو أوم

R2 : مقاومة (١/٤) واط (١) ميغا أوم

R3 : مقاومة (١/٤) واط (١) كيلو أوم

R4,R5 : مقاومة (١/٤) واط (100)كيلو أوم

R6 : مقاومة متغيرة (٥٠ كيلو أوم)

C1,C2 : مكثف عدسي (٠,٠١ميكرو فاراد)

: C3مكثف عدسي (١٠٠) بيكو فاراد

: C4مكثف عدسي (٠,٠٤٧) ميكرو فاراد

: C5مكثف عدسي (٠,١) ميكرو فاراد

: C6مكثف كيميائي (٣,٣ (ميكرو فاراد (١٦) فولت

: C7مكثف كيميائي (١,٥ (ميكرو فاراد (١٦) فولت

: Q1ترانزستور 2N2222 (NPN) أو مكافئه (٢N3904)

: Q2ترانزستور 2N2907 (PNP)

: Q3ترانزستور ضوئي(NPN)

: D1 ديود ١N914

IC1 : LM308
IC2 : LM567

: LED1 ليد أشعة تحت الحمراء

حاكمة : حاكمة (٦) فولت

: S1مفتاح لحظي

: B1بطارية (٣) فولت (بطاريتان (1,5) فولت على التسلسل)

MISC

لوحة فيبير ، قواعد للدارات المتكاملة، ذراع للمقاومة المتغيرة، حامل بطارية

❖ مبدأ العمل:

توضع دارات المرسل والمستقبل على الأسوار الخارجية للمنزل والمداخل الأساسية. وعندما يتم قطع الضوء بين المرسل والمستقبل (ضوء الأشعة تحت الحمراء) ترسل إشارة من المستقبل إلى المتحكم الصغري (المايكروكونترولر) ومنه إلى مدخل ال com على الحاسب عن طريق دارة الموائمة Max 232

٩- محركات التيار المستمر : DC Motors

محركات التيار المستمر ذات طبيعة حثية (تحريض بالحقل المغناطيسي) وبالتالي تستحق أن تعامل بنفس طريقة الريليه

في مشروعنا الحالي يمكن للميكروكونترولر أن يتحكم في تغيير سرعة محرك التيار المستمر ، لذا الغرض يتم التنفيذ بوحدة (موديول PWM) والميكروكونترولر 16F628A بمذبذب 4MHz .

كود البرنامج كما يلى:

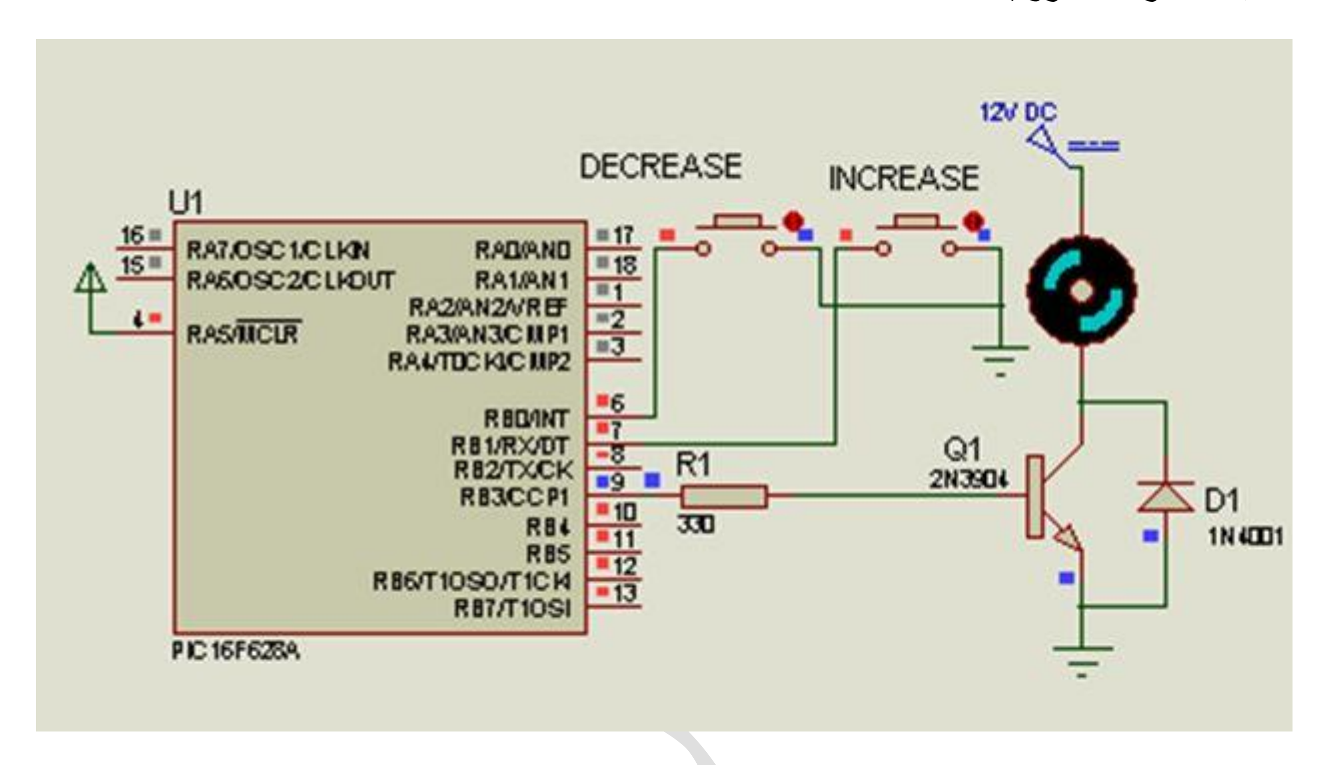
```
Code:
void main( void )
{
unsigned short CU=0;
          OPTION_REG = 0;            //Activate the pull-up resistors.
          PWM1_Init( 500 );   //PWM module starts at 500 Hz
          PWM1_Set_Duty(CU);
          PWM1_Start();
while(1)        //Infinite Loop.
          {
                    //loop to increment PWM  when pressed RB1.
while( Button( &PORTB, 1, 10, 0 ) )
                    {
                              CU++; if( CU==0 )CU=255;
                              PWM1_Set_Duty(CU);
delay_ms(10);
                    }
                    //loop decrement PWM  when pressing RB0
while( Button( &PORTB, 0, 10, 0 ) )
                    {
                              CU--; if( CU==255 )CU=0;
                              PWM1_Set_Duty(CU);
delay_ms(10);
                    }
          }
}
```

```
void main( void )
{
    unsigned short CU=0;
    OPTION_REG = 0;            //Activate the pull-up resistors.
    PWM1_Init( 500 );          //PWM module starts at 500 Hz
    PWM1_Set_Duty(CU);
    PWM1_Start();
    while(1)               //Infinite Loop.
    {
        //loop to increment PWM  when pressed RB1.
        while( Button( &PORTB, 1, 10, 0 ) )
        {
            CU++; if( CU==0 )CU=255;
            PWM1_Set_Duty(CU);
            delay_ms(10);
        }
        //loop decrement PWM  when pressing RB0
        while( Button( &PORTB, 0, 10, 0 ) )
        {
            CU--; if( CU==255 )CU=0;
            PWM1_Set_Duty(CU);
            delay_ms(10);
        }
    }
}
```

❖ التجهيزات المطلوبة للتصميم:

16F628A, BUTTON, RES, 2N3904, 1N4001, MOTOR ACTIVE

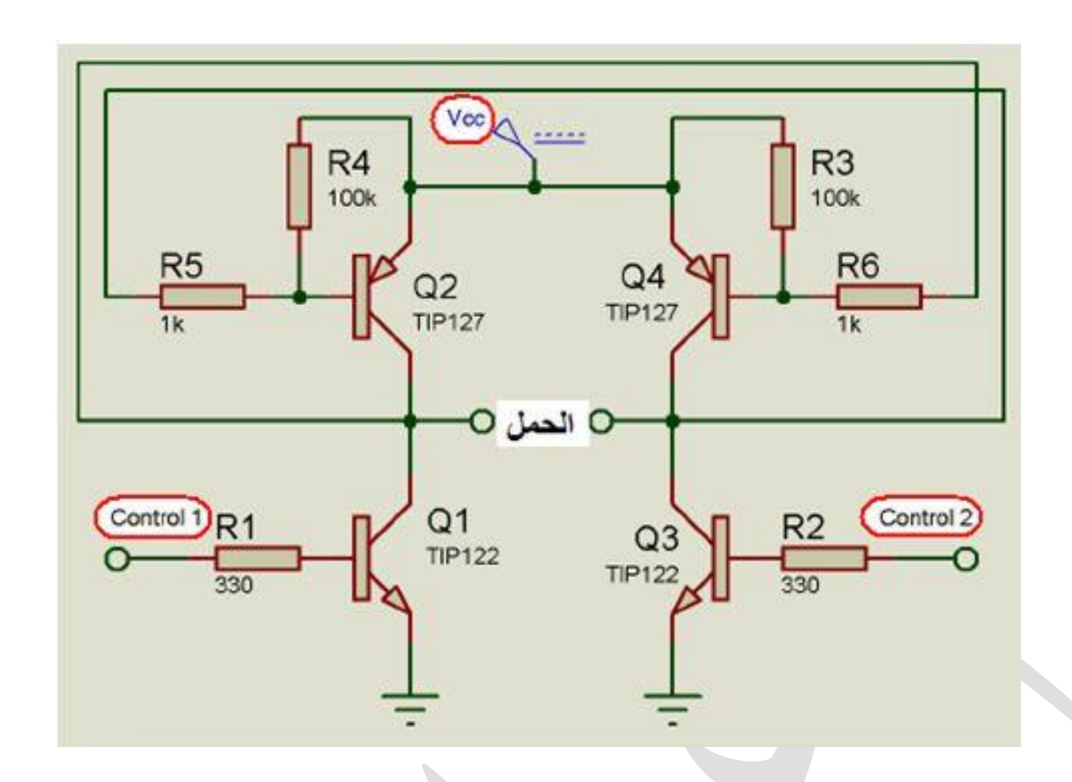
– لبناء الدارة الالكترونية:



تبعا لشدة التيار المطلوب يمكن تغيير الترانزستور لتحسين أداء النظام . الترانزستور 2N3904يمكنه التحكم في تيار حتى 200m Amps والترانزستور TIP31يمكنه التحكم في تيار حتى 3 amps تبعا لنظام العمل ومنحنى الخواص .

القنطرة(الجسر) إتش H Bridge:

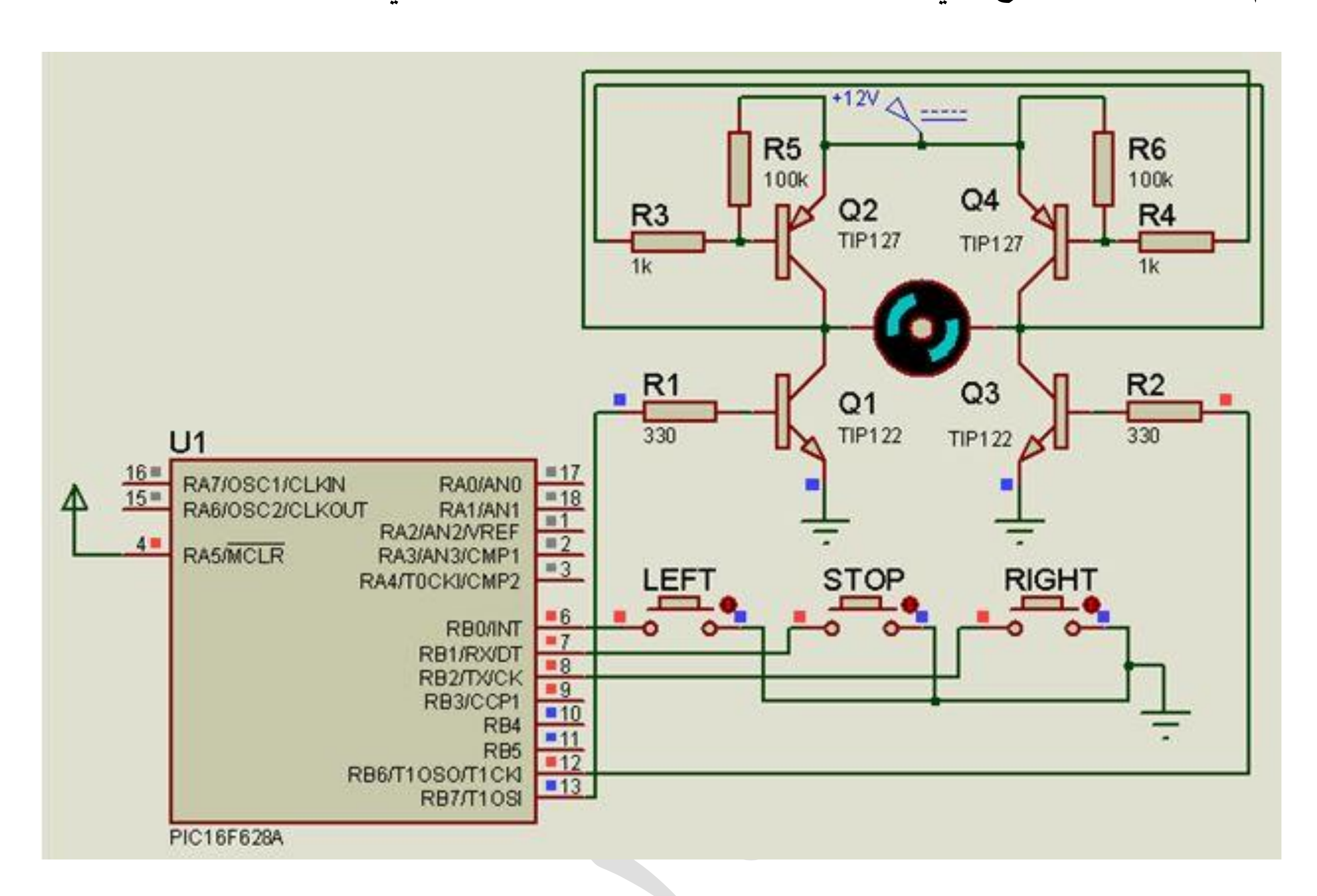
القنطرة H هي ترتيب (تنظيم) من الترانزستورات والذى يسمح بعكس قطبية الحمل الكهربائي عند توفير مصدر جهد مفرد (أحادي القطبية . single source) يمكن بناء القنطرة H باستخدام ترانزستورات منفصلة discreteأو باستخدام موديولات متكاملة والمتاحة تجاريا، هذا الترتيب يتضمن إشارتين تحكم للتمكين من اختيار القطبية الموجبة والقطبية السالبة . الشكل التالي يبين الدائرة الكهربائية للترتيب المنفصل:



لتفعيل القطبية الموجبة يتم تفعيل خط التحكم الأول control 1 بينما يتم فصل خط التحكم الثاني control 2 ولتفعيل القطبية السالبة يتم عكس حالة أطراف التحكم، مع الوضع في الاعتبار أن خطى التحكم لن يتم أبد التحكم فيهما في نفس الوقت، حيث أن هذا الفعل ينتج عنه دائرة قصر short بين الجهد الموجب Vcc والأرضي، مما يتسبب في أضرار جسيمة بالترانزستورات.

```
1   void main( void )
2   {
                                                    تفعيل مقاومات الرفع الداخلية
3       OPTION_REG = 0; //Activate the pull-up resistors.
4       //Configuring port B  as output and input .
5       TRISB = 0x0F;
6       PORTB = 0;
7
8       while(1) //Infinite Loop.
9       {                                   حلقة اكتشاف الضغط على مفتاح الدوران جهة اليسار
10              //Loop for detecting the rotation to the left button .
11              while( Button( &PORTB, 0, 10, 0 ) )
12              {                 فصل خطى التحكم لإيقاف المحرك بغرض الحماية من الدوان العكسى
13                      PORTB = 0;      //Turn off the control pins.
14                      PORTB.F7 = 1;  //Activates the control pin 1.
                                        تفعيل خط التحكم الأول للدوران جهة اليسار
15              }
16                                          حلقة اكتشاف الضغط على مفتاح الإيقاف
17              //Loop for detecting the stop button.
18              while( Button( &PORTB, 1, 10, 0 ) )
19              {
20                      PORTB = 0;      //Turn off the control pins.
21              }
22                                    حلقة اكتشاف الضغط على مفتاح الدوران جهة اليمين
23              //Loop for detecting the rotation  to the right button.
24              while( Button( &PORTB, 2, 10, 0 ) )
25              {                 فصل خطى التحكم لإيقاف المحرك بغرض الحماية من الدوان العكسى
26                      PORTB = 0;      //Turn off the control pins.
27                      PORTB.F6 = 1;  //Activates the control pin 2.
28              }                       تفعيل خط التحكم الثانى للدوران جهة اليمين
29      }
30  }
```

يتم تنفيذ المحاكاة ببرنامج دلفي بعد بناء الدائرة الالكترونية المبينة بالشكل الآتي:



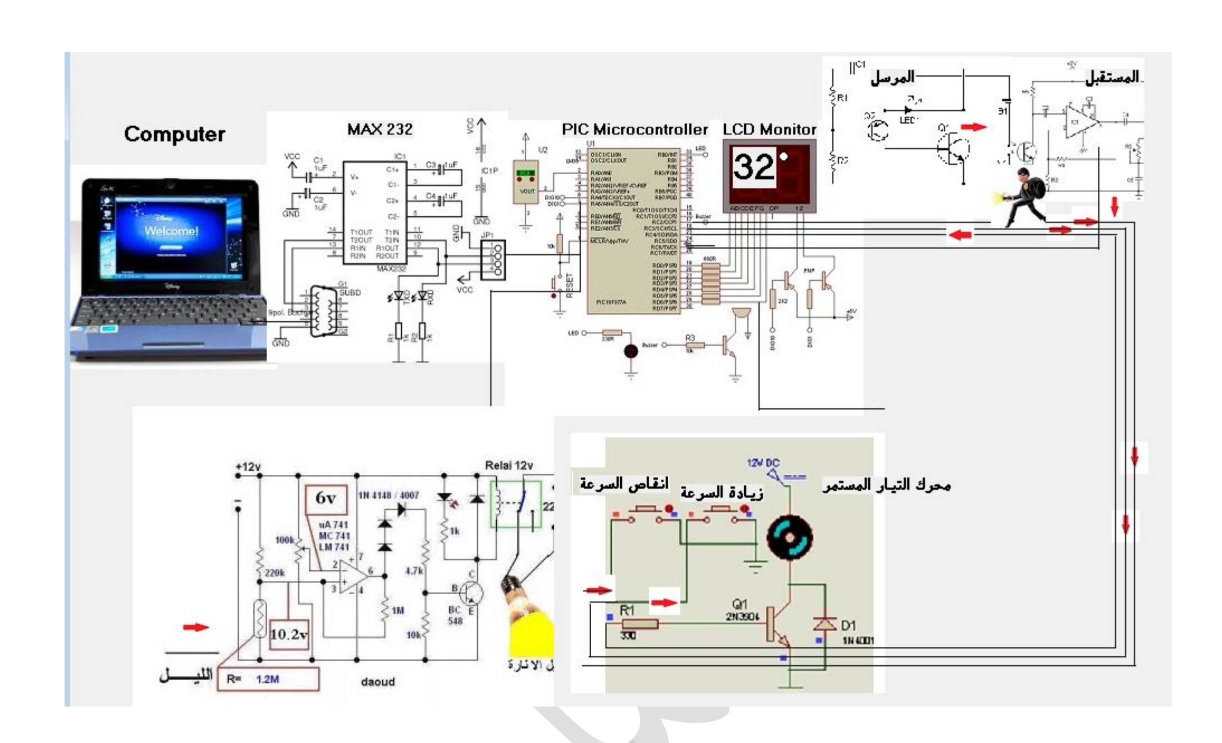
يتم ربط هذه الدارة (دارة محرك التيار المستمر) بالبوابة الرئيسية ليتم التحكم الآلي عن طريق المايكروكونترولر وواجهة برنامج الدلفي بفتح واغلاق البوابة.

❖ واجهات المشروع:

❖ **الواجهة الرئيسية:**



❖ واجهة التطبيق :



❖ البرنامج:

```
unit Unit2;


interface


uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls, Buttons;


type
  TForm2 = class(TForm)
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    Image4: TImage;
    Image5: TImage;
    Image6: TImage;
    Shape1: TShape;
    Shape2: TShape;
    Shape3: TShape;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
```

```
Label4: TLabel;
Shape4: TShape;
Shape5: TShape;
Label5: TLabel;
Shape6: TShape;
Shape7: TShape;
Shape8: TShape;
Shape9: TShape;
Image8: TImage;
Shape11: TShape;
Shape12: TShape;
Label6: TLabel;
Label7: TLabel;
Timer1: TTimer;
Timer2: TTimer;
Timer3: TTimer;
Timer4: TTimer;
Image9: TImage;
Bevel1: TBevel;
BitBtn1: TBitBtn;
Image7: TImage;
Image11: TImage;
Image12: TImage;
Image13: TImage;
Image14: TImage;
```

```
Image15: TImage;
Image16: TImage;
Image17: TImage;
Image18: TImage;
Image19: TImage;
Image20: TImage;
Image21: TImage;
Image22: TImage;
Image23: TImage;
Image24: TImage;
Shape13: TShape;
Shape14: TShape;
Image25: TImage;
Image26: TImage;
Image27: TImage;
Shape15: TShape;
Timer5: TTimer;
Image30: TImage;
Label9: TLabel;
Shape10: TShape;
Shape17: TShape;
Shape18: TShape;
Shape19: TShape;
Shape20: TShape;
Shape21: TShape;
```

```
    Shape22: TShape;
    Shape23: TShape;
    Shape24: TShape;
    Shape25: TShape;
    Shape26: TShape;
    Shape27: TShape;
    Image10: TImage;
    Label8: TLabel;
    Image28: TImage;
    Label10: TLabel;
    Label11: TLabel;
    Image29: TImage;
    Image31: TImage;
    Image32: TImage;
    Image33: TImage;
    Image34: TImage;
    Timer6: TTimer;
    Image35: TImage;
procedure Timer1Timer(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Timer3Timer(Sender: TObject);
procedure Timer5Timer(Sender: TObject);
procedure Timer4Timer(Sender: TObject);
private
```

```
{ Private declarations }
public
{ Public declarations }
end;


var
  Form2: TForm2;


implementation
uses unit1;
{$R *.dfm}


procedure TForm2.Timer1Timer(Sender: TObject);


begin
if x<36  then
begin
x:=strtoint(label5.Caption);
x:=x+1;
 label5.Caption:=inttostr(x);
  image9.Visible:=false;
  image25.Visible:=false;
end
else
```

```
begin
   image9.Visible:=true;
   image25.Visible:=true;
x:=x-1;
   label5.Caption:=inttostr(x);
end;
end;



procedure TForm2.BitBtn1Click(Sender: TObject);
begin
form2.Close;
form1.Show;
timer1.Enabled:=false;
timer2.Enabled:=false;
timer3.Enabled:=false;
timer4.Enabled:=false;
timer5.Enabled:=false;
image22.Visible:=false;
image15.Visible:=false;
image16.Visible:=false;
image10.Visible:=false;
end;
```

```
procedure TForm2.Timer2Timer(Sender: TObject);
begin
if Image26.Visible=true then
begin
 Image26.Visible:=false;
 Image17.Visible:=true;
 Image16.Visible:=false;
 Image15.Visible:=true;
 Image22.Visible:=false;
 Image21.Visible:=true;
end
else
begin
   Image26.Visible:=true;
   Image17.Visible:=false;
   Image16.Visible:=true;
 Image15.Visible:=false;
 Image22.Visible:=true;
 Image21.Visible:=false;
end;


end;


procedure TForm2.Timer3Timer(Sender: TObject);
```

```
begin
if Image4.Visible=true then
begin
 Image4.Visible:=false;
 Image8.Visible:=true;
 image27.Visible:=true;
 image18.Visible:=false;
 image19.Visible:=false;
end
else
begin
 Image4.Visible:=true;
 Image8.Visible:=false;
 image27.Visible:=false;
 image18.Visible:=true;
 image19.Visible:=true;
end
end;


procedure TForm2.Timer5Timer(Sender: TObject);
begin
if Image34.Visible=true then
begin
 Image34.Visible:=false;
 Image32.Visible:=true;
```

```
 Image31.Visible:=false;
 Image29.Visible:=true;
 Image22.Visible:=false;
 Image21.Visible:=true;
end
else
begin
   Image34.Visible:=true;
   Image32.Visible:=false;
   Image31.Visible:=true;
 Image29.Visible:=false;
 Image35.Visible:=true;
 Image21.Visible:=false;
end;


end;


procedure TForm2.Timer4Timer(Sender: TObject);
begin
if image7.Visible=true then
begin
   image28.Visible:=false;
   image7.Visible:=false;
   image11.Visible:=true;
```

```
  image12.Visible:=true;
  image13.Visible:=false;
  image14.Visible:=false;
  label8.Visible:=true;
end
else
begin
   image7.Visible:=true;
   image28.Visible:=true;
  image11.Visible:=false;
  image12.Visible:=false;
  image13.Visible:=true;
  image14.Visible:=true;
  label8.Visible:=false;
end;


end;


end.
```

❖ المصطلحات العملية:

المصطلح	معنى المصطلح
MICROCONTROLOR	المتحكم الصغري
MAX232	دارة الملائمة
COM	منفذ حاسوب
LM35	حساس الحرارة
DC Motor	محرك مستمر
Start Bit	بت بداية
Data Bits	بتات المعطيات
Parity Bit	بت الإزدواجية
Stop Bits	بتات التوقف
S-board V2.0	لوحة ملائمة المنفذ التسلسلي

❖ المراجع:

المراجع العلمية:

١- برمجيات متقدمة في نظم التحكم للدكتور عبدالرحمن الحسين منشورات جامعة حلب

٢- هندسة الكترونيات والتحكم للدكتور خلف العبدالله .منشورات جامعة حلب

٣- الكترونيات الرقمية للدكتور عمر حلي منشورات جامعة حلب

٤- برمجيات في نظم الحاسوب للدكتور محمد العاني منشورات جامعة دمشق

١-http://WWW.MICROCHIP.COM

٢-www.dATASHEET PIC1.com

٣-http://www.mikroe.com

٤- http://www.labcenter.com.uk